An incremental approach to building a mathematical expert out of software

Richard Fateman
University of California
Berkeley

Abstract: Build an Expert

- Pick a class of problems, and then simulate, on a computer, an expert human solving that problem.
- Or take all the solved problems from all the books, and use Google to find answers.

- *Why don't these ideas work?*
- *What ideas might work?*
- We address these questions based on some 40 years of progress in symbolic math programs, including some impressive strides.
- Q&A
- As time permits, a case study of building an internet expert on symbolic integration.

Outline

- Aiming High: The Mathematical Golem
- Aiming Lower: Principle of Least Surprise
- Math, Formalism, Algorithms, Data, Putting it together

Creating Smart Programs

- With apologies to http://golem.plush.org/faq
- What is a Golem and why is it like a computer system?

What is a Golem?

- A golem is an artificial creature in Jewish Cabalistic folklore
  - It is an automaton or robot
  - It is made of clay and dust
  - It is endowed with life by mystical means (secret inscriptions, incantations, etc.)
- Note that clay and dust consists of carbon, silicon, etc. --- the building blocks of computers.

An example

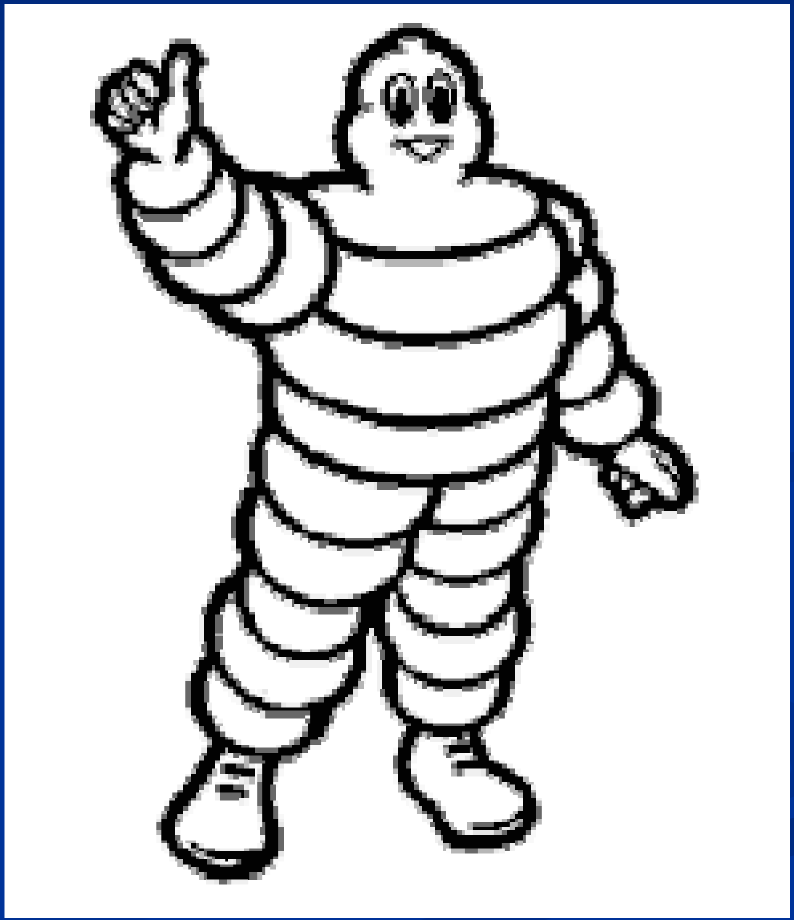
- The Golem of Prague

Not a Golem

- Bibendum,

the Michelin Man,

is made of tires, not clay.



What’s so great about a Golem?

- A Golem is subservient to its creator, though in legend it can become uncontrollable, and grows more powerful from day to day.
- In some depictions, this clay creature takes commands from a piece of paper inserted in its mouth. Of course this was before floppy disks and CD-ROMS.

What's so great about a Golem? (part II)

- Golems follow the instructions of their master without question.
- Extremely strong and practically indestructible.
- Good at carrying out menial tasks and are handy around the house.
- They don't feel pain.
- They don't speak.
- Very low upkeep.
- Golems traditionally used for defense systems.

Frequently Asked Questions about golems {www} / [computers]

- Q. I've followed all of the directions to make a golem [write a program] correctly but it still won't work. What did I do wrong?
- A. Creation of a golem [or program] is generally reserved for those learned, wise and righteous persons. If you can't successfully create the golem, either you have made a slight mistake in pronunciation or you have not achieved the proper state of holiness.

- Q. Will the golem kill me?
- A. This is entirely within the realm of possibility. Golems seem more uncontrollable the longer they have been in existence. Thus, the longer you use the golem, the more likely it will rampage. [On the other hand, maintaining a program merely sucks the life blood out of you.]

- Q. Why was Golem so fixated on the Ring?
- A. That's ``Gollum'' not ``Golem.''

What does this have to do with Mathematics / Axiom Computer System?

- Joel Moses (MIT, 1967 landmark PhD thesis on *symbolic indefinite integration* by computer) was dedicated

*To the descendants of the Maharal **

who are endeavoring to build a Golem.

**Rabbi Loew of Prague, b. 1525*

Can we build a Golem to do math?

- What do we mean by “do math”??
  - Do what a professional mathematician does?
  - Do what a scientist does, (applied math)?
  - Read/comment on a journal paper?
  - Do what a student does?
    - What level?
    - How well?

Start at the top

- We know that (some) people, so-called mathematicians, can "solve mathematical problems."
  - Carry out routine mechanical algebraic transformations
  - Imagine, Theorize, Conjecture, novel items.
  - Prove, Disprove conjectures.
  - Conduct discourse with other mathematicians on math topics.
  - Decide a result is novel, esthetically pleasing, worthy of dissemination. (Publish, get tenure, etc)
- Is it absurd to think that we can simulate on a computer *some* of these activities?
- How will we know if we have succeeded?
- Do we have a benchmark to test for a computerized "world's best mathematician"?

This is Really Too Hard (Impossible?)

- Mathematicians disagree among themselves about what they do. It's the Artificial Intelligence challenge: encode
  - ***INTENT***
  - ***WORLD VIEW***
  - ***CONTEXT / ASSUMPTIONS***
  - ***Varied and conflicting*** language and formalisms
- No benchmarks: If a computer earned an "on-line" degree in math, would you be surprised?
- Compare to chess…

When did computers succeed in chess?

- By displaying the board; identifying legal moves?
- Beating 99.9% of players, or
- Beating the (human) world champion? Once?

What if we start from the bottom…. Could we claim success

- If we can simulate some set of tools?
  - Arithmetic, completely accurate to arbitrary precision.
  - Totally routine proof-following program (verifying theorems)
- If we can use a computer to *teach* math.
  - Better than current teachers??

Maybe we can creep up to a higher level

- If a computer is viewed as a valuable assistant for some humans who use math, can we say it “does math”?
- What if it tutors students / does homework in math?
- e.g. www.calc101.com (really!)
  - “**Buy a password to boost your algebra and calculus grades…**
  - Get all the steps for indefinite integrals.
  - Get all the steps for determinants and matrix inverses.
  - Get all the steps for systems of linear equations.
  - Get a complete analysis of the graph of your function.”

Is this a fair freshman calculus homework or test problem? Compute:

$$\int x^5 \sin x \; dx$$

Is this a fair freshman calculus homework or test problem? Compute:

$$\int x^5 \sin x \; dx = \left(5\,x^4 - 60\,x^2 + 120\right)\, \sin x + \left(-x^5 + 20\,x^3 - 120\,x\right)\, \cos x$$

How? Integrate by parts, repeatedly. Is this too hard for a student problem? Is this a good exercise or just drudgery? How much of this task is a valuable human skill? WWW.Calc101 can do this and show all steps. Does this prove that Calc101 "does math"?

This is about where we are now

- Computers do some math, especially tedious kinds of math, better than humans.
- Some mathematicians/programmers use “raw” computing for various reasons. (e.g. in C or assembler!)
- Some mathematicians use sophisticated software (e.g. Axiom) to formulate and carry out “experiments”.
  - Testing conjectures
  - Trying numerical calculations

- Programs Construct tables, diagrams, graphs, typeset papers.
- Some **math/computer scientists** explore the limits of what can be done mechanically; and how fast. How to communicate math.
- Algorithms, languages, data structures, interfaces.

Where do we go from here?

- We want to extend the reach of computer programs: more ambitious tasks.
- We want the programs to be correct (Even more: we want to be able to show they are correct.)
- We want to make them easier to use for research, education, applications.
- *** we will pursue this line for the remainder of our time.

Principle of Least Surprise (POLS)

- This is an informal heuristic used for the design of programming languages, user interfaces, and probably much else.
- It says that if there is an otherwise arbitrary choice point A or B in the design of a system, and most humans would expect the system to do A, then the system should do A.

Axiom, Computers, Math and POLS

- Some people confuse Mathematics with Arithmetic.
- Some people confuse Arithmetic with “What Computers Do.”
- Thus, by the Principle of Least Surprise, Computers should do Math.
- Although a *mathematician* would be surprised to find this the case!

Is this an unsurprising program? Google's Calculator

- Google has a calculator. You can type into a search string (3+4)*5

- What will Google do? Return (3+4)*5=35
- Now that you know this, by POLS you might expect it to do many other things.

More from Google's Calculator

- 100!/99!=
- 100!/99!=100
- 170!/169!=
- 170!/169!=170
- 171!/170!=
- <random search stuff>

POLS fails: why?

- The maximum "IEEE double float" number $1.7976931348623\ldots \times 10^{308}$ is a consequence of arithmetic performance on most computers. This particular computer-geeky limit has no *mathematical* importance, but it means:
- $170! = 7.25741562\ldots \times 10^{306}$ is smaller than this and is legal.
- $171!$ is $1.241018070217\ldots \times 10^{309}$ which is "too big."

Distrust grows

- The *computer program breaks*, even though the *mathematics is nearly the same.*
- Mathematics being a science devoted to generalization, allows us to believe the relation

  $n! = n \times (n\text{-}1)!$

  Even if we do not know the value of *n*.

  Even if *n* is not an integer.

Distrust grows

- Other bogus computations often provided in common languages:
  - For some numbers, n and n+1 are indistinguishable because of roundoff.
  - Because of integer overflow, calculations near the top of the range overflow to very negative numbers. On a 32-bit computer, if you are given $h=2^{31}-1$, computing $h+1$ may get you -2^{32}.

Computer Algebra Systems can Restore Trust

- The integers do not stop at 2^{32}-1, or 2^{64}-1 etc
- The axioms for groups, rings, fields, etc can be observed (finitely represented at least)
- The real numbers like π and *e* are not strangers.
- Exactitude holds a place of honor; approximation is used with caution.

Computer Algebra Systems can Restore Trust: Caveats

- Every program has bugs. Some have design errors too.
- Explicit representation of a number (in a CAS or on a blackboard) can run into some physical limit. This might be related to the number of bits in your computer's memory.
- If the known universe (in 2005, estimated at 30 billion light years across) were packed densely with atoms each about 10^{-10} meter across, we would accommodate about 10^{73} atoms.
- Would we be unhappy if a CAS can only count up to about $2 \times 10^{646,456,887}$?

Clearly positional notation helps ☺.

Where does that leave us?

- The advantage of thinking with the aid of a CAS is that you have the traditional advantages of a computer,
  - Able to follow directions
  - Fast and Cheap
  - Nice to use for email, library access, downloading music, etc.
- Yet you work with representations of many more numbers and even symbols like *x, n, π.*
- Overall fewer surprises, less computer-geeky. Gets those examples like 171!/170! = 171 and even n!/(n-1)!=n **correct. Math is more in charge.**

Math is not good enough

- Much more informality is used in conveying math than is commonly recognized.
- Conveying math to an *idiot savant* requires a level of explicitness often missing from usual communication.
- New language constructs (perhaps new to mathematicians and to computer scientists) must be considered.

A concern for forms

- A mathematics text may not distinguish between x^2-y^2 and (x-y)(x+y).
- after all, they both represent the same mathematical function.
- But there is an essential *computational* difference, in that they have different FORMS. Only one is factored.

How Forms Matter

- What is the determinant of this matrix?

$$\begin{pmatrix} x-y & 1 \\ x^2-y^2 & y+x \end{pmatrix}$$

- It tends to matter if it is zero or not.
- $(x-y)\times(x+y)-1\times(x^2-y^2)$ must therefore be simplified to zero. Programs implement transformations on forms.
- How many programs? What must they do? And there is more trouble: if we have sin(x), cos(x) etc. And is x “real” or does it matter?

We want simple, convenient forms

- Which is “better”?

$$\frac{x^{20}-1}{x-1}$$

OR

$$x^{19}+x^{18}+x^{17}+x^{16}+x^{15}+x^{14}+x^{13}+x^{12}$$
$$+x^{11}+x^{10}+x^{9}+x^{8}+x^{7}+x^{6}+x^{5}+x^{4}+x^{3}+x^{2}+x+1$$

How Algorithms Matter

- Students are rarely taught algorithms
  - Expected to absorb principles from examples
  - Theorems may be gratuitously non-constructive
- Teachers assign problems to minimize computation.
- (Heuristic: the numbers in this exercise are unwieldy → an error was committed earlier)
- Not all "real world" problems come out neatly.

How Data Structures Matter

- Encode a polynomial $3t^{10}+4$
  - $\langle\, 3\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 4 \,\rangle$
  - $\langle\, \langle\, 10\ 3 \rangle\ \langle\, 0\ 4 \,\rangle \,\rangle$
- Many variants possible, with significant differences when you operate on them, or even store them in memory.

Algorithms + Data

- The algorithms used by a CAS may be quite different from those used by calculus students, and may in fact be quite unknown to the instructors.
- Devising (preferably efficient) algorithms for additional areas of inquiry are part of the CAS research program.
- A selection of active areas:
  - Solution of polynomial systems
  - Integration of special functions (Hypergeometrics Fns)
  - Solution of ODEs
  - Combinatorial identities
  - Group theory
  - Geometric proof / Cylindrical Algebraic Decomposition

Among CAS, Axiom's Strength

- Compared to other CAS, Axiom's designers were more devoted to making a clean foundation for building additional capabilities.
- This design permeates the programming language used "internally" as well as the view given to the casual Axiom user. Every object has associated domain information.

Heuristics, Search, Hackery

- By no means is Axiom's approach the only way.
- What if Google were extended so that instead of just computing (3+4)*5, it just searched the world of the internet?
- Could this do math?
- Uh, what if it could do all freshman calculus textbook problems, and more?
- Could we incorporate all math in "Ask Dr. Math"? Clearly not *all*. But "what is a rhombus?" … 99%?

Conclusion

- Maybe math expertise is too ephemeral to incorporate in a finite collection of algorithms in one place.
- Maybe the World Wide Web can include
  - library-style collections of mathematics, suitably encoded
  - Programmed “agents” that are skilled in certain areas could be maintained by distributed groups of (qualified?) humans. These agents could evolve (compete) to be world-class experts.
- Today’s CAS may be basic algorithms plus a “friendly” set of customized user interfaces to different agents for different purposes.

What about a web site that does symbolic integration?

- Three plausible ways

1. Try to run a computer algorithm
   - convert every input to an expression in a differential field for which there is an integration procedure.
   - Returns a formal integral or a result that the answer does not exist in closed form in terms of elementary functions.

2. Run some heuristic pattern matching on the input and try to find a solution, or a transformation on the way to a solution.

3. Look up the question in a big table.

- Tilu does 3. It uses a mixture of hash-coding and tree search, plus a little solving for coefficients during match.
- Other sites (Calc101, WRI, try 2, then 1)

Tilu, a Table of Integrals LookUp Web site

- Begun circa 1996, with Ted Einwohner.
- 800 general patterns, a few hundred extras.
- 250 hits a day, now (300,000 saved examples!)
- Accessible from Macintosh Graphing Calculator (one key to integrate!)
- Written entirely in Common Lisp (web server too)
- All patterns are Lisp data
- New patterns can be added in real time while Tilu is running.

v Help

torte.cs.berkeley.edu:8010/tilu

Search

okmarks My Yahoo! Maps ACL6.2 Index ACL 6.2 TOC

Choose from the following:

Back to TILU

Mathematica/Common Lisp Grammar

Get a wrong answer?

About TILU

Comments/Questions

TILU Table of Integrals Look Up

(version 6.0 AllegroServe - frames)

Welcome to TILU, a web server that looks up integrals in a reference table.

If you have never used this page before, or if you need some review on the syntax for describing your integral, please be sure to check in the frame on the left. Once you have figured out how to ask your question, type or paste the integrand in the big box below:

If TILU just seems to be waiting, hit the stop button. This is an experimental server and is not always up.

If the default settings for options are acceptable with you, click on the activate button. Otherwise, change the entries in the boxes before clicking.

Before you send us some mathematics, you must tell us the language you are using. The choices include Common Lisp, Mathematica, and a third, we call ``Some random language I make up as I go along." Our log of old entries suggests most people are best off if they use the last of these. In any case, choose your language, and choose the limits of integration. To simplify your input task, we assume that you are always integrating with respect to the variable x.

We really would like to hear comments!

Select Language | Indefinite Integral

Activate integral request

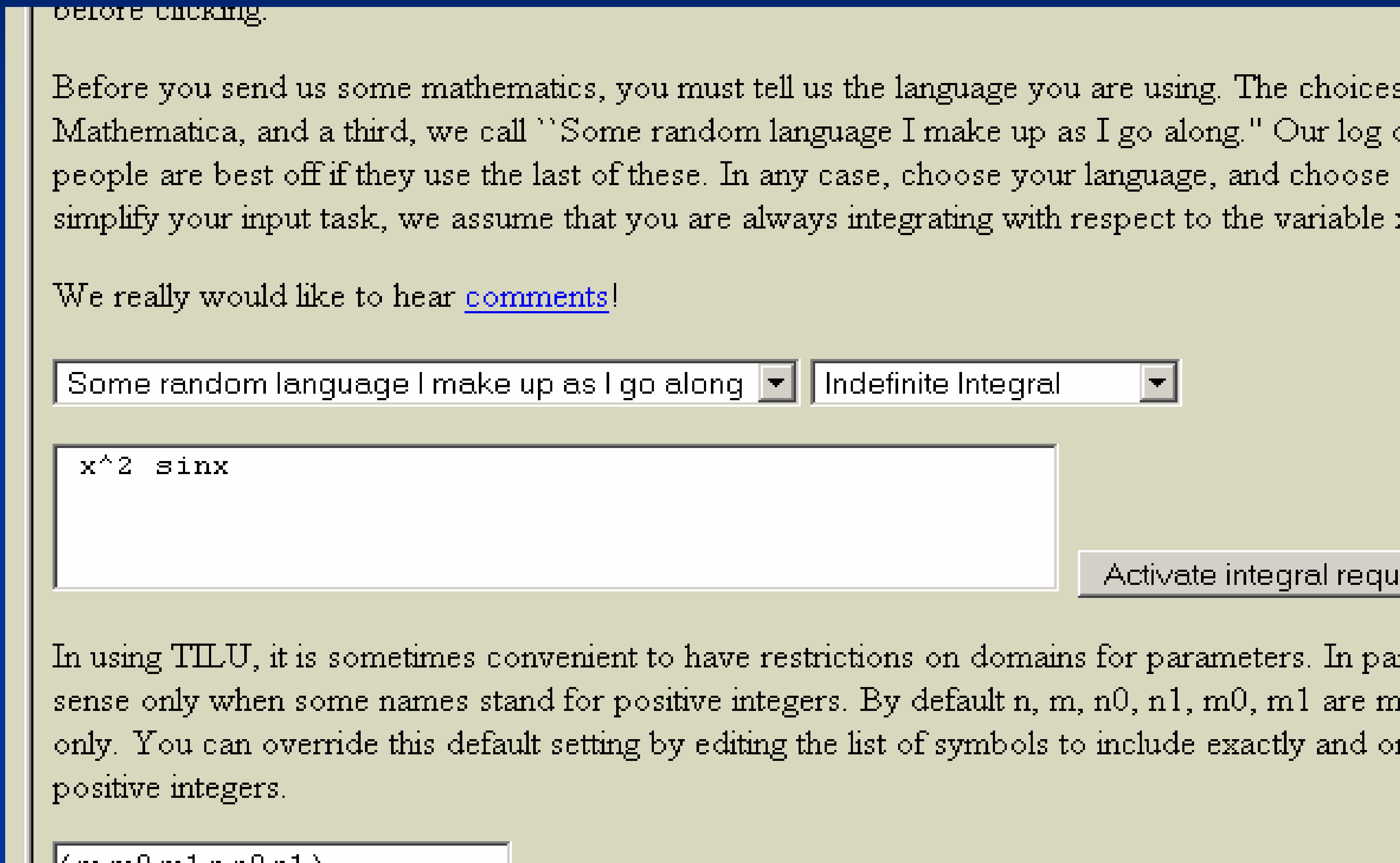
In using TILU, it is sometimes convenient to have restrictions on domains for parameters. In particular, some integrals make sense only when some names stand for positive integers. By default n, m, n0, n1, m0, m1 are meant to stand for positive integers only. You can override this default setting by editing the list of symbols to include exactly and only those that you want to match positive integers.

(m m0 m1 n n0 n1)

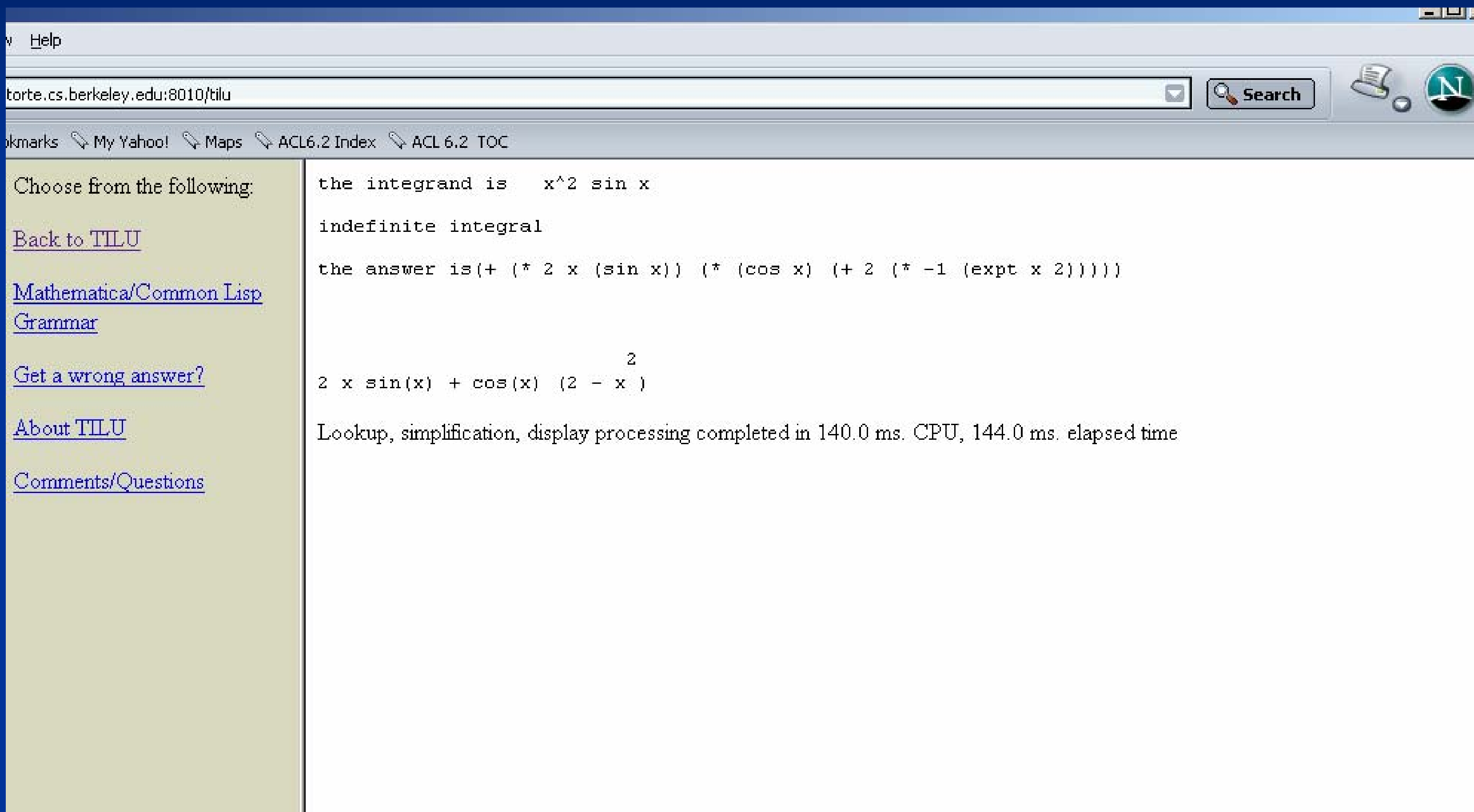
All other parameters are assumed to stand for arbitrary real numbers.

Prior to January, 1998 we processed about 7000 integrals. As of November, 2004 we are processing about 220 queries a day, about 40 percent from the integrate key on the Macintosh graphing calculator, and the rest through web browsers. Tilu runs interchangeably on Windows NT, XP, Linux, Sun Solaris, HP UNIX, and other systems running Allegro Common Lisp 6.2.

Closeup of input panel



Solution page



Closeup

```
the integrand is   x^2 sin x

indefinite integral

the answer is(+ (* 2 x (sin x)) (* (cos x) (+ 2 (* -1 (expt x 2))))


                                      2
2 x sin(x) + cos(x) (2 - x )
```

Lookup, simplification, display processing completed in 140.0 ms. CPU, 144.0 ms. elapse

About the answer

- We provide a Lisp “text” result.
- We provide a “simulated 2-D” display, too.
- We could provide MathML or GIF for typeset output, but that’s a different project.
- Sometimes there are multiple pattern hits; the result includes them all.
- Note that sinx was changed to sin(x).

What was hard about Tilu?

- Visitors have difficulty typing in well-formed formulas. We had to come up with a very generous error-correcting parser.
- Apparently no one reads any instructions on the internet so everything has to either
  - run by default or
  - stop unless required data is supplied (forced).

What could improve Tilu?

- Interface with (say) Google calculator
- Expansion to a smart “math knowledge” agent on the internet.
  - Algorithms for when patterns do not match (careful to limit time expenditure)
  - Solving other applied math problems e.g. ODEs
  - Combination with DLMF (digital library of math functions from NIST)